汕头市众利电子科技有限公司

产品说明书

ZL1604A1

联系地址：广东省汕头市潮阳区平北外经工业村 1-4 楼
联系电话：0754-8724000
传真号码：0754-8729633
电子邮箱：zhlcm@zhonglilcm.com
公司网址：www.zhonglilcm.com

目 录

概 述

人们对液晶显示器并不陌生,最常见的有如计算器、电子表、数字万用表、电子游戏机等,显示的主要是数字、专用符号和固定图形,因为是属段式显示,显示内容就无法多变。

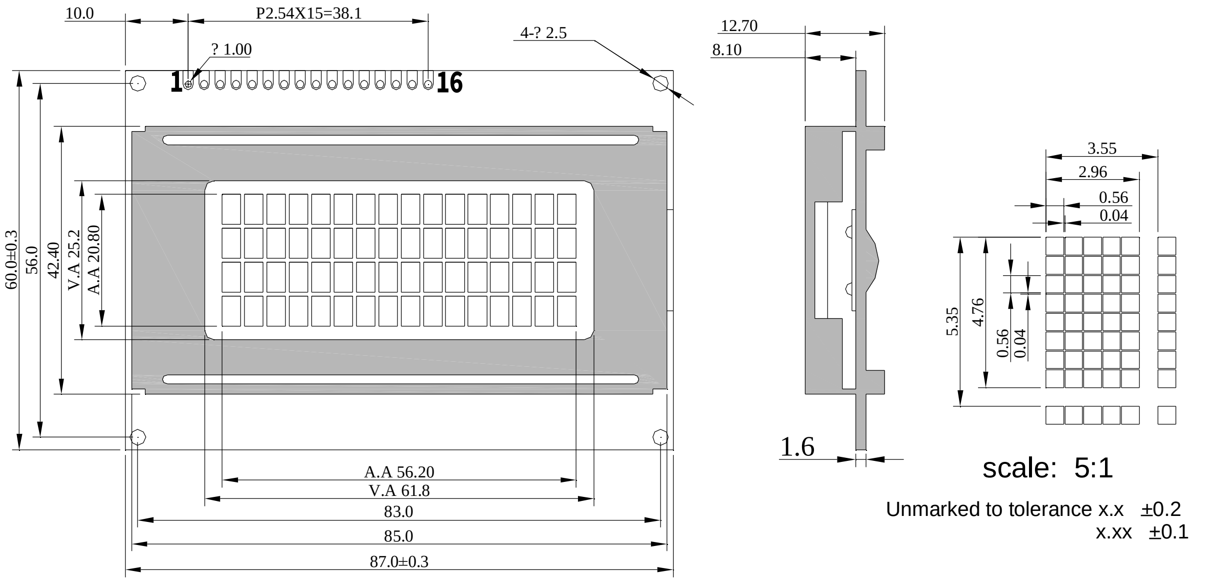
随着大量电子仪器、设备的智能化,并且普遍地采用人机交互方式,需要能够显示更为丰富的信息和通用性较强的显示器,而点阵式 LCD 显示器能够满足这些要求,同时用大规模专用集成电路作为点阵 LCD 控制驱动,使用者仅仅直接送入数据和指令可实现所需的显示。这种由 LCD 板、PCB 板、控制驱动电路组成的单元叫做点阵液晶显示模块(DOT MATRIC LCD MODULE)。

广东汕头市众利电子科技有限公司是液晶显示器专业生产厂家，以其雄厚的力量，先进的生产设备及工艺，已开发生产出一系列的 LCD 点阵模块（字符型和图形型）。本手册着重介绍字符型模块的使用方法。

1. 字符型模块的性能

重量轻:≤100g;
体积小:≤11mm 厚;
功耗低:10 - 15 mw;
显示内容:192 种字符(5×7 点字型);
32 种字符(5×10 点字型);
可自编 8(5×7)或 4(5×10)种字符;
指令功能强:可组合成各种输入、显示、移位方式以满足不同的要求;
口简单方便:可与 8 位微处理器或微控制器相联;
工作温度宽:-20℃ - 70℃;
可靠性高:寿命为 50,000 小时(25℃)

2:机械尺寸:



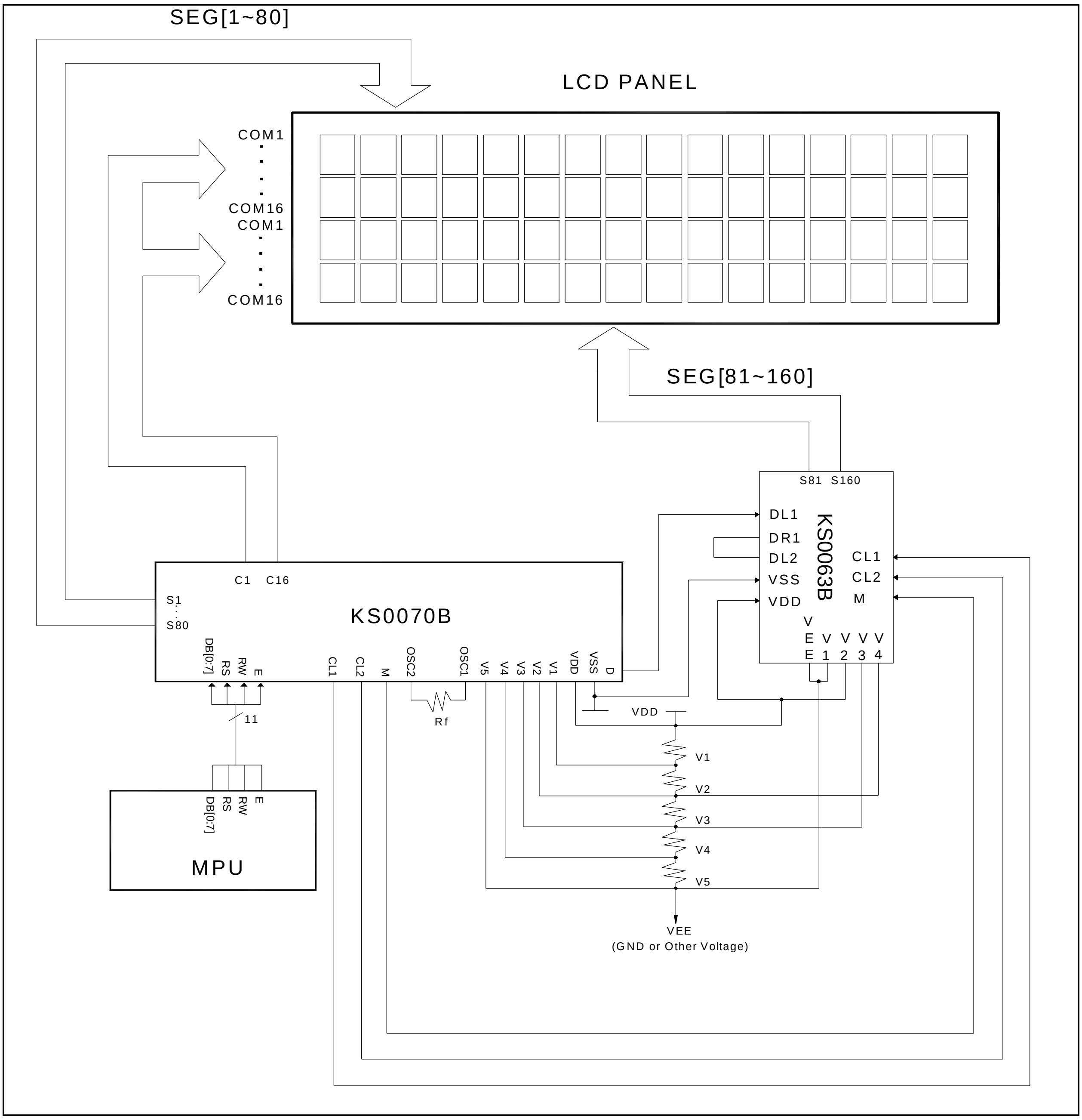
3.基本原理

3.1 液晶板

在液晶板上排列着若干 **5**×**7** 或 **5**×**10** 点阵的字符显示位,每个显示位可显示 **1** 个字符,从规格上分为每行 **8,16,20,24,32,40** 位,有一行,两行及四行三类。

3.2 工作电路

图 **1** 是字符型模块的电路框图,它由 **KS0070B,**及几个电阻电容组成。**KS0063** 是扩展显示字符位用的



***Note**

1/16 duty, 1/5 bias

$V1 = V_{DD} - V_{LCD}/5$
$V2 = V_{DD} - 2V_{LCD}/5$
$V3 = V_{DD} - 3V_{LCD}/5$
$V4 = V_{DD} - 4V_{LCD}/5$
$V5 = V_{DD} - V_{LCD}$

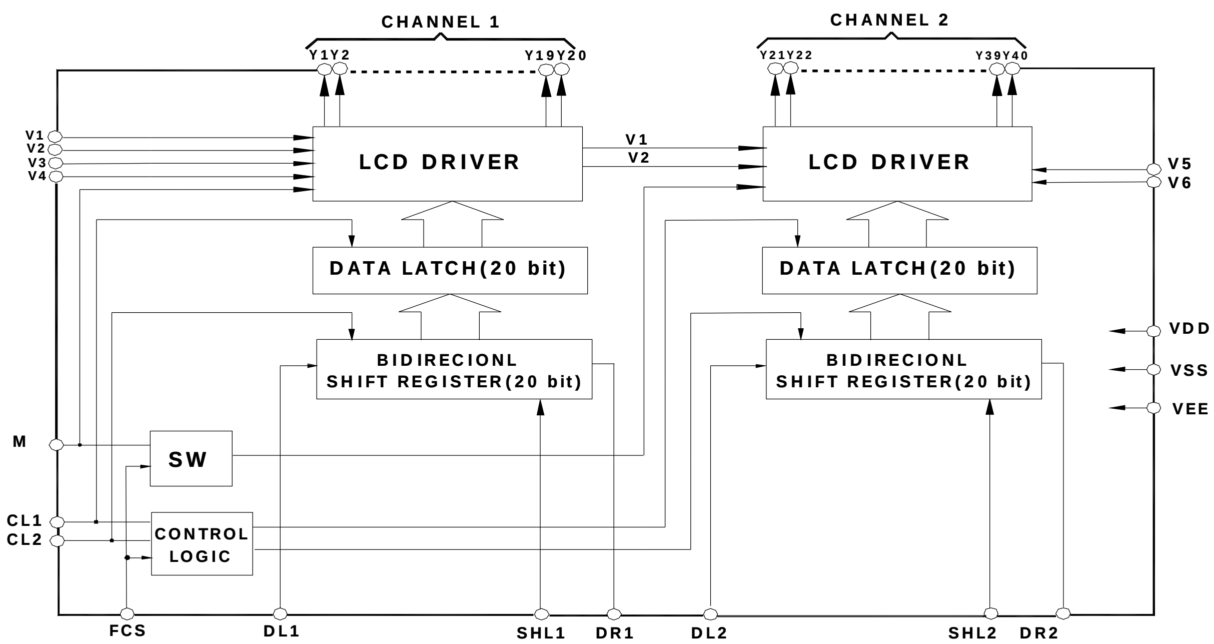
接口方面,有 8 条数据线,三条控制线。可与微处理器或微控制器相连,通过送入数据和指令,就可使模块正常工作.

3. 3 LCD 驱动器和控制器

3.3a LCD 驱动器(KS0063)

KS0063 是用低功耗 CMOS 技术制造的大规模 LCD 驱动 IC。它既可当行驱动用,也可当列驱动用,由 40 × 2Bits 二进制移位寄存器, 40 × 2Bits 数据锁存器和 40 × 2Bits 驱动器组成(如图 3 所示)。

BLOCK DIAGRAM



3 KS0063 function Block diagram

▲ 功能(FUNCTION)

a 40 通道点阵 LCD 驱动;
b 可选择当作行驱动或列驱动;
c 输入/输出信号:输出,能产生 40×2 个 LCD 驱动波形;输入,接受控制器送出的串行数据和控制信号,偏压(V1∽V6);

▲ 特性(FEATTURES)

a 显示驱动偏压比:静态∽1/5;
b 电源电压:+5V±10%;

c 显示驱动电源(VDD-VEE)为-5V;
d CMOS 处理;
e 100 引脚,塑封;软封;

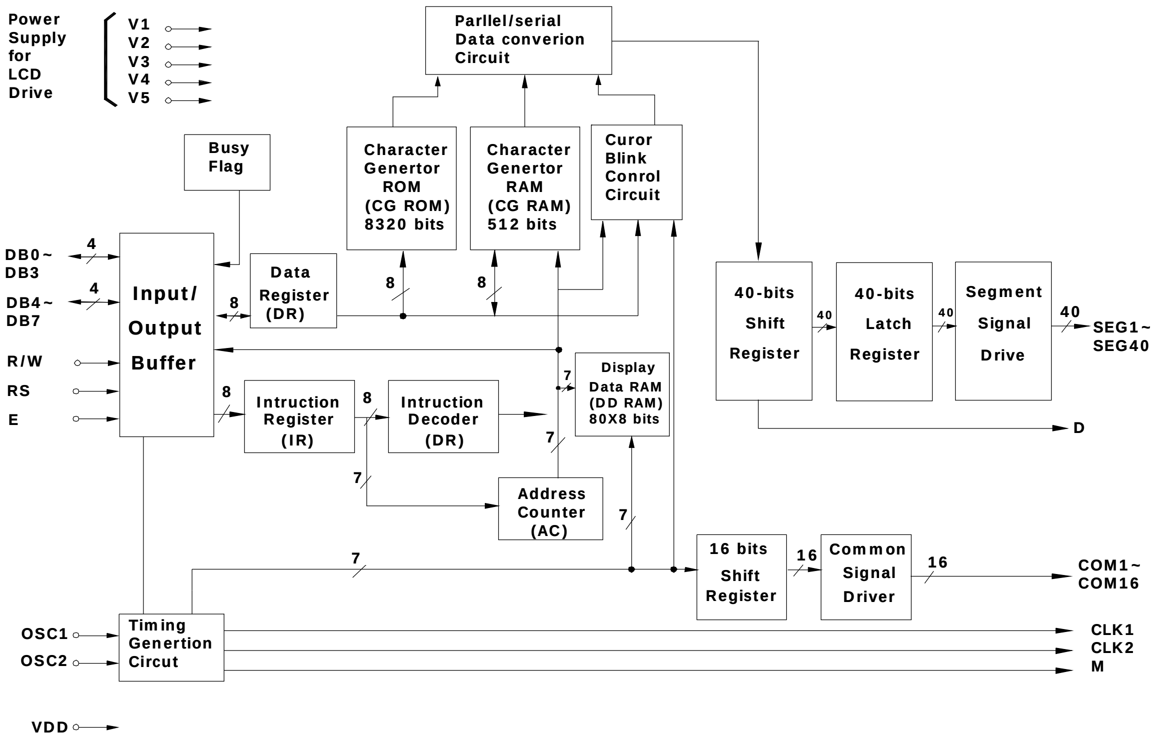
3.3b LCD 控制器 KS0070B

见图 4,(KS0070B)是用低功耗 CMOS 技术制造的大规模点阵 LCD 控制器(兼带驱动器),和 4Bit/8Bit 微处理器相连,它能使点阵 LCD 显示大小英文字母,数字和符号。应用 H(KS0070B),用户能用少量元件可组成一个完整点阵 LCD 系统。

▲ 特性

a 容易和 4Bit/8Bit MPU 相连;
b 可选择 5×7 或 5×10 点阵字符;
c 显示数据 RAM 容量:80×8Bit (80 字符);
d 字符发生器 ROM 能提供用户所需字符库或标准库;
字库容量:192 个字符(5×7 点字型);
32 个字符(5×10 点字型);
e DDRAM 和 CGRAM 都能从 MPU 读取数据;(DDRAM 为显示缓冲区;CGRAM 为可自编数据区)
f 输出信号:16 个行扫描信号(common signal),
80 个列扫描信号(segment signal);
g 电源复位电路;
h 显示占空比:1/8duty(1 Line, 5×7dots + Cursor);
1/11duty(1 Line, 5×10dots + Cursor);
1/16duty(2 Line, 5×7dots + Cursor);
通常为 1/16 duty.
I 振荡电路;
J 指令:11 种;
K 128 引脚,塑封,软封。

BLOCK DIAGRAM



4 KS0070B function block diagram

3.4 背光参数

字符模块通常带 LED 背光板。它的性能参数如下:

工作温度:-20∽+70° C;

存储温度:-40∽+80° C;

背光板可显示绿色,红色,黄色,兰色和白色。背光一般为黄绿色，也可为客户设计为其他颜色，但价格较绿色昂贵许多。

正常工作电流（如：模块 1602）为：60∽80mA（若要达到白天效果需 85mA);

工作电压：4.2∽5V;

正常工作条件下，LED 可连续点亮 5 万小时;

4.技术参数

4.1 极限参数

名 称	符 号	标 准 值			单 位
		MIN	TYPE	MAX	
电路电源	V_{DD} - V_{SS}	-0.3		7.0	V
LCD 驱动电压	V_{DD} - V_{EE}	V_{DD} - 13.5		V_{DD} + 0.3	V
输入电压	V_{IN}	-0.3		V_{DD} + 0.3	V
静电电压		-	-	100	V
工作温度		-20		+70	° C
储存温度		-30		+80	° C

4.2 电参数

名 称	符 号	测 试 条 件	标 准 值			单位
			MIN	TYPE	MAX	
输入高电平	V_{IH}	-	2.2		V_{DD}	V
输入低电平	V_{IL}	-	-0.3		0.6	V
输出高电平	V_{OH}	I_{OH} = 0.2mA	2.4		-	V
输出低电平	V_{OL}	I_{OL} = 1.2mA	-		0.4	V
工作电流	I_{DD}	V_{DD} = 5.0V		2.0		mA
液晶驱动电压	V_{DD} - V_{EE}	Ta = 0° C		4.9		V
		Ta = 25° C		4.7		
		Ta = 50° C		4.5		

4.3 光学参数

名 称	符 号	测 试 条 件	标 准 值			单位
			MIN	TYPE	MAX	
视角	θ1-θ2	K≥2.0	2.5			deg
对比度	K	θ2=20° ψ=0°	8.0			
上升时间	τr	θ2=20° V_D = 4.1V		100	150	ms
下降时间	τd	θ2=20° V_D = 4.1V		150	200	ms

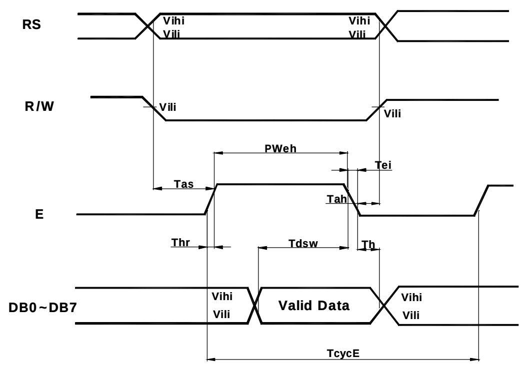
5.时序特性

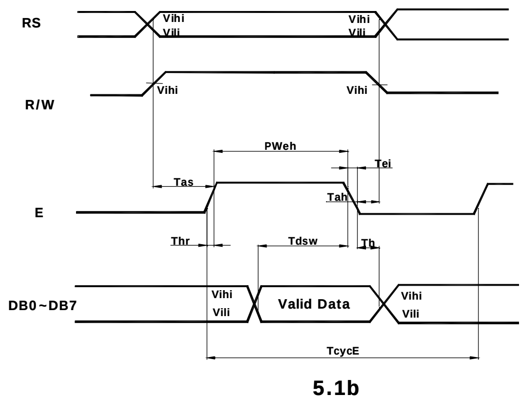
5.1 读写时序

TIMING CHART						时 序
项 目	符 号	测试条件	标 准 值			单位
			MIN	TYPE	MAX	
允许时间周期	T_{CYCE}	5.1a 5.1b	1000			ns
允许脉冲宽度,高电平	PW_{EH}		450	--	--	ns
允许上升和下降时间	t_{Er} t_{Ef}		--	--	25	ns
地址建立时间	t_{AS}		140	--	--	ns
数据延迟时间	t_{DDR}		--	--	320	ns
数据建立时间	t_{DSW}		195	--	--	ns
数据保持时间	t_H		10	--	--	ns
DATA HOLD TIME	t_{DHR}		20	--	--	ns
地址保持时间	t_{AH}		10	--	--	ns

※ Vcc = 5.0V±5%,Ta = 25° C

5.1a WRITE OPERATION (写操作)　　　　5.1b READ OPEERATION (读操作)

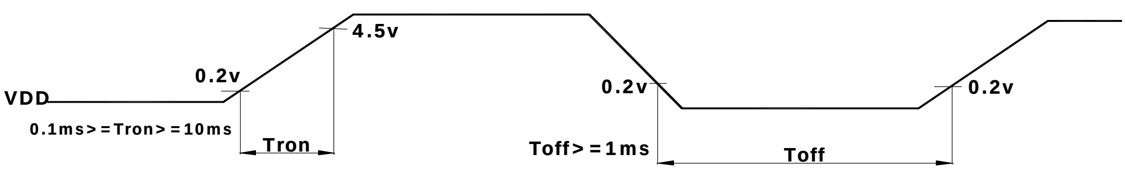




5.1a　　　　5.1b

5.2 内部 RESET 电路对电源的要求

	符 号	测 试 条 件	MIN	MAX	单 位
电源上升时间	T_{RON}	5.2a	0.1	10	ms
电源下降时间	T_{OFF}	5.2a	1.0	--	ms



NOTE Toff stipulates the time of power OFF for power supply instantaneous dip Or when power supply repeats ON and OFF.

5.2a

1.如电源不能满足 **RESET** 电路的要求,需要用指令程序进行初始化。
2.所有符号只要字母相同(不分大小写),所代表的意义就相同。

6.引脚和指令功能

6.1 模块引脚功能

引 线 号	符 号	名 称	功 能
1	Vss	接地	0V
2	VDD	电路电源	5V±10%
3	VEE	液晶驱动电压	保证 VDD-VEE=4.5∽5V 电压差
4	RS	寄存器选择信号	H:数据寄存器 L:指令寄存器
5	R/W	读/写信号	H:读 L:写
6	E	片选信号	下降沿触发,锁存数据
7 | 14	DB0 | DB7	数据线	数据传输
15	LEDA	背光源正极	------
16	LEDK	背光源负极	------

6.2 寄存器选择功能

RS	R/W	操 作
0	0	指令寄存器(IR)写入
0	1	忙标志和地址计数器读出
1	0	数据寄存器(DR)写入
1	1	数据寄存器读出

备注:忙标志为"1"时,表明正在进行内部操作,此时不能输入指令或数据,要等内部操作结束,忙标志为"0"时。

6.3 指令功能

格式:RS R/W DB7 DB6 DB5 DB4 DB3 DB2 DB1 DB0

共**11**种指令:清除,返回,输入方式设置,显示开关,控制,移位,功能设置,CGRAM地址设置,DDRAM地址设置,读忙标志,写数据到CG/DDRAM,读数据由CG/DDRAM。

指 令 表

指令名称	指令码										说明	执行周期
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		FCP=250KHZ
清屏	L	L	L	L	L	L	L	L	L	H	清除屏幕,置AC为零	1.64ms
返回	L	L	L	L	L	L	L	L	H	X	设DDRAM地址为零,显示回原位,DDRAM内容不变	1.64ms
输入方式设置	L	L	L	L	L	L	L	H	I/D	S	设光标移动方向并指定整体显示是否移动	40us
显示开关控制	L	L	L	L	L	L	H	D	C	B	设整体显示开关(D),光标开关(C),及光标位的字符闪耀(B)	40us
移位	L	L	L	L	L	H	S/C	R/L	X	X	移动光标或整体显示,同时不改变DDRAM内容	40us
功能设置	L	L	L	L	H	DL	N	F	X	X	设接口数据位数(DL),显示行数(L),及字型(F)	40us
CGRAM地址设置	L	L	L	H	ACG						设CGRAM地址,设置后CGRAM数据被发送和接收	40us
DDRAM地址设置	L	L	H	ADD							设DDRAM地址,设置后DDRAM数据被发送和接收	40us
读忙信号(BF)及地址计数器	L	H	BF	AC							读忙信号位(BF)判断内部操作正在执行并读地址计数器内容	0us
写数据CG/DD RAM	H	L	写数据								写数据到CG或DDRAM	40us Tadd=6ns
读数据由CG/DD RAM	H	H	读数据								读数据由CG或DDRAM	40us Tadd=6ns
	I/D 1:增量方式, 0:减量方式 S 1:移位 S/C 1:显示移位,0:光标移位 R/L 1:右移,0:左移 DL 1:8位,0:4位 N 1:2行,0:1行 F 1:5×10,0: 5×7 BF 1:内部操作,0:接收指令 RS :寄存器选择 R/W :读/写										DDRAM : 显示数据RAM CGRAM: 字符生成RAM AC: 用于DD和CGRAM地址的地址计数器	执行周期随主频率改变而改变 例如:当Fosc或fcp为270KHZ 40 us × 250/270= 37 us

6.4 字符库及对应关系

6.4.1 显示位与DD RAM 地址的对应关系

显示位序号		1	2	3	4	5	……………………	40
DD RAM 地址(HEX)	第一行	00	01	02	03	04	…………………….	27
	第二行	40	41	42	43	44	…………………….	67

7.1 标准字符库

(KS0070B-00)


Upper 4bit
Lower 4bit
LLLL LLHL LLHH LHLL LHLH LHHL LHHH HLLL HLLH HLHL HLHH HHLL HHLH HHHL HHHH
CG RAM (1)
LLLL
LLLH (2)
LLHL (3)
LLHH (4)
LHLL (5)
LHLH (6)
LHHL (7)
LHHH (8)
HLLL (1)
HLLH (2)
HLHL (3)
HLHH (4)
HHLL (5)
HHLH (6)
HHHL (7)
HHHH (8)

7.2 自编字库(CGRAM)

字符码(**DDRAM DATA**),**CGRAM ADDRESS** 与自编字形(**CGRAM DATA**)之间关系如表 **3** 和表 **4** 所示:

Table 3. 5×7 dots character pattern(indicate "SST")

DD RAM Data (Character Codes)		CG RAM Address		CG RAM Data (Character Pattern)
7 6 5 4 3 2 1 0 MSB LSB		5 4 3 2 1 0 MSB LSB		7 6 5 4 3 2 1 0 MSB LSB
L L L L X L L L		L L L L L L L L H L H L L H H H L L H L H H H L H H H		X X X L H H H H H L L L L H L L L L ↓ L H H H L L L L L H L L L L H H H H H L L L L L L
L L L L X L L H		L L H L L L L L H L H L L H H H L L H L H H H L H H H		X X X L H H H H H L L L L H L L L L ↓ L H H H L L L L L H L L L L H H H H H L L L L L L
L L L L X L H L		L H L L L L L L H L H L L H H H L L H L H H H L H H H		X X X H H H H H L L H L L L L H L L ↓ L L H L L L L H L L L L H L L L L H L L L L H L L
⋮		⋮		⋮
L L L L X H H H		H H H L L L L L H L H L L H H H L L H L H H H L H H H		X X X ↓

X:DON'T Care

备注:1.字符码的高 4 位为 0000 时它的低 3 位对应于第 1-8 个(000 - 111)自编字形;
2.字符码的 0 - 2 位对应于 CGRAM 地址的 3 - 5 位;
3.自编字形的列位置对应于 CGRAM DATA 的 0 - 4 位,行位置对应于 CGRAM ADDRESS 的 0 - 2 位;
4.X 代表无效位;
5.H 代表显示位;

Table 4. 5×10 dots character pattern(indicate "KⅡ")

DD RAM Data (Character Codes) 7 6 5 4 3 2 1 0 MSB LSB		CG RAM Address 5 4 3 2 1 0 MSB LSB		CG RAM Data (Character Pattern) 7 6 5 4 3 2 1 0 MSB LSB
		L L L L		X X X L L L L L
		L L L H		L L L L L
		L L H L		H L L L H
		L L L L H H		↓ H L L H L
		L H L L		H L H L L
L L L L X L L X		L H L H		H H L L H
		L H H L		H L H L L
		L H H H		L L L H L
		H L L L		L L L L H
		H L L H		L L L L L
		H L H L		L L L L L
		L L H L H H		X X X X X
		H H L L		
		H H L H		
		H H H L		
		H H H H		
		L L L L		X X X L L L L L
		L L L H		H H H H H
		L L H L		L H L H L
		L H L L H H		↓ L H L H L
		L H L L		L H L H L
L L L L X L H L		L H L H		L H L H L
		L H H L		L H L H L
		L H H H		L H L H L
		H L L L		H H H H H
		H L L H		L L L L L
		H L H L		L L L L L
		L H H L H H		X X X X X
		H H L L		
		H H L H		
		H H H L		
		H H H H		
⋮		⋮		⋮
		L L L L		X X X X X X X X
		L L L H		
		L L H L		
		H H L L H H		
		L H L L		
L L L L X H H X		L H L H		
		L H H L		
		L H H H		
		H L L L		
		H L L H		
		H L H L		
		H H H L H H		
		H H L L		
		H H L H		
		H H H L		
		H H H H		

X:DON'T Care

8．初始化方法

用户所编的显示程序,开始必须进行初始化,否则模块无法正常显示,下面介绍两种初始化方法;

8.1 利用内部复位电路进行初始化

如果电路电源能满足图 5.2 所示的条件的话,就可实行初始化,下面指令是在初始化过程中执行的。

(1)清屏**(DISPLAY CLEAR);**

(2)功能设置**(FUNCTION SET);**

DL = 1: 8Bit 接口数据**;**

N = 0: 1行显示**; F = 0:5×7dot** 字形**;**

(3)显示开/关控制**(DISPLAY ON/OFF CONTROL)**

D = 0: 显示关**; C = 0:** 光标关**; B = 0:** 消隐关

(4)输入方式设置**(ENTRY MODE SET)**

I/D = 1:(增量**):S = 0:** 无移位**:**

8.2 软件复位

如果电路电源不能满足复位电路的要求的话,那么初始化就要用软件来实现,过程如下:

八位接口初始化流程图

电 源 开

↓

VDD 上升到 4.5V 后等待>15

↓

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	1	1	×	×	×	×

↓ 等待**>4.1ms**

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	1	1	×	×	×	×

↓ 等待**>100us**

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	1	1	×	×	×	×

↓ 检查忙标志或延时 **40us**

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	1	1	N	**F**	×	×

↓ 检查忙标志或延时 **40us**

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	1	**0**	×	×

↓ 检查忙标志或延时 **40 us**

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	**0**	**0**	**1**

↓ 检查忙标志或延时 **1.64**

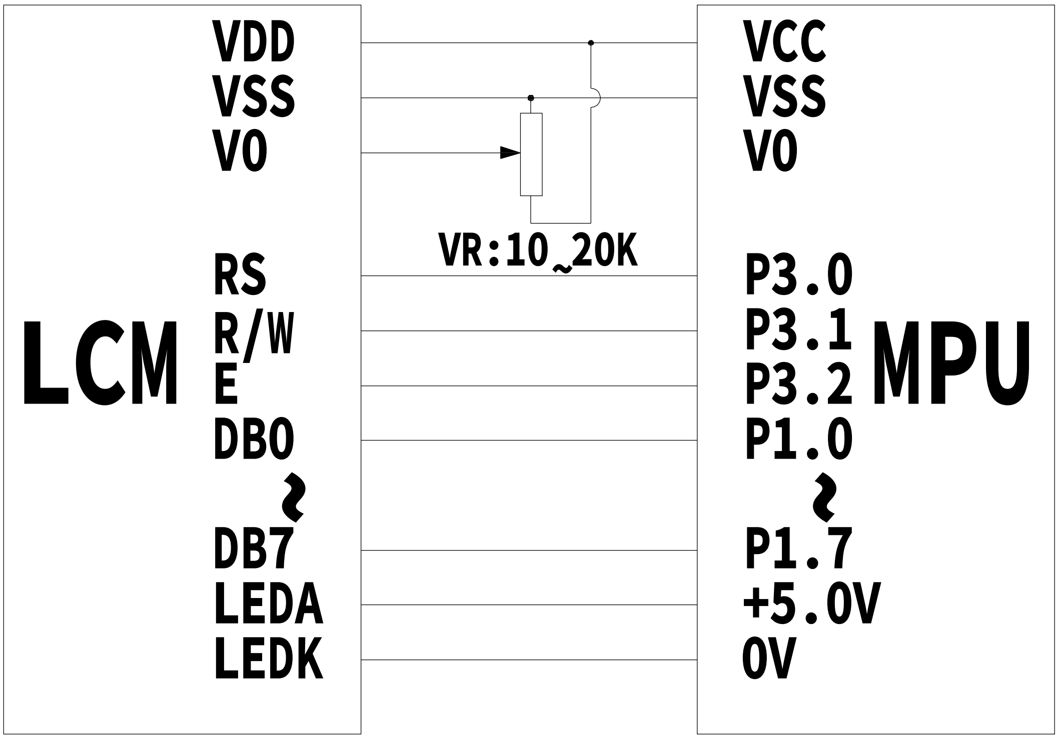
RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	**1**	**I/D**	**S**

↓ 检查忙标志或延时 **40**

初 始 化 结 束

9. 程序举例：

以下程序为利用 P3 口控制,8BIT 数据线传输;端口连接控制;接口图如下:




```
;   控制器 KS0070B 或 SPLC780   显示程序
RS      BIT     P3.0
RW      BIT     P3.1
E       BIT     P3.2
SW      EQU     P3.3
        ORG     0000H
        AJMP    START
        ORG     0030H
START:
        SETB    RS
        CLR     E
        CLR     RW
        lcall   delay_10ms
        MOV     A,#01H              ;clear screen
        ACALL   WINS
        lcall   delay_10ms
        MOV     A,#02H              ;return home
        ACALL   WINS
        lcall   delay_10ms
        MOV     A,#06H              ;entry mode set
        LCALL   WINS
        lcall   delay_10ms
        MOV     A,#0EH              ;display ON/OFF
```

```
        ACALL   WINS
        LCALL   DELAY_2MS
        MOV     A,#38H                  ;function set
        LCALL   WINS
        LCALL   DELAY_2MS
        MOV     A,#18H                  ;Cursor or display shift
        LCALL   WINS
        LCALL   DELAY_2MS
MAIN:
;-----第一屏--以下显示网格内容----------------------------
        MOV     A,#40H          ;set  CG  RAM  ADDRESS
        mov     p1,A
        LCALL   WINS
        MOV     A,#0AH
        LCALL   WDATA
        MOV     A,#15H
        LCALL   WDATA
        MOV     A,#0AH
        LCALL   WDATA
        MOV     A,#15H
        LCALL   WDATA
        MOV     A,#0AH
        LCALL   WDATA
        MOV     A,#15H
        LCALL   WDATA
        MOV     A,#0AH
        LCALL   WDATA
        MOV     A,#15H
        LCALL   WDATA
;--------------------------------------------
        MOV     A,#01H
        LCALL   WINS
        mov     a,#80h
        lcall   wins
        mov     r1,#40H         ;产生网格点16进制数量,40H即64D
ddA:
        mov     a,#00h
        lcall   wdata
        djnz    r1,ddA
        MOV     A,#0C0H
        LCALL   WINS
        MOV     R1,#20H
DWW:
        MOV     A,#00H
        LCALL   WDATA
        DJNZ    R1,DWW
        LCALL  WT               ;调用换屏子程序,换屏用
;****************************************************
```

```
;-----第二屏--以下 IC 字库代码内容----------------------------
        MOV     A,#01H
        LCALL   WINS
        LCALL   DELAY_2MS
        MOV     A,#80H
        LCALL   WINS
        MOV     R1,#32D          ;液晶屏显示字符之总列数
        MOV     DPTR,#TAB1
CHAR1:
        CLR     A
        MOVC    A,@A+DPTR
        LCALL   WDATA
        INC     DPTR
        DJNZ    R1,CHAR1
        MOV     R1,#32D         ;液晶屏显示字符之总列数
        MOV     A,#0C0H
        LCALL   WINS
CHAR2:
        CLR     A
        MOVC    A,@A+DPTR
        LCALL   WDATA
        INC     DPTR
        DJNZ    R1,CHAR2
;        LCALL   DELAY_1S
;*************************************************
   LCALL WT               ;调用换屏子程序,换屏用
;-----第三屏--以下 IC 字库代码内容----------------------------
        MOV     A,#01H
        LCALL   WINS
        LCALL   DELAY_2MS
        MOV     A,#80H
        LCALL   WINS
        MOV     R1,#32D        ;液晶屏显示字符之总列数
        MOV     DPTR,#TAB2
CHAR3:
        CLR     A
        MOVC    A,@A+DPTR
        LCALL   WDATA
        INC     DPTR
        DJNZ    R1,CHAR3
        MOV     R1,#32D       ;液晶屏显示字符之总列数
        MOV     A,#0C0H
        LCALL   WINS
CHAR4:
        CLR     A
        MOVC    A,@A+DPTR
        LCALL   WDATA
        INC     DPTR
```

```
        DJNZ     R1,CHAR4
;        LCALL    DELAY_1S
;*************************************************
        LCALL   WT          ;调用换屏子程序,换屏用
        AJMP     MAIN
;------------------------------------------------
WINS:
        CLR      RS
WDATA:

        ACALL    DELAY_2MS
        MOV      P1,A
        SETB     E
        NOP
        NOP
        CLR      E
        SETB     RS
        RET
;---------------------------
WT:     SETB    SW
        MOV     C     ,SW
        LCALL     DELAY_40us
        JC      WT
        LCALL   DELAY_2ms
        SETB    SW
        MOV     C     ,SW
        LCALL     DELAY_40us
        JNC     WT
        RET
;---------------------------

DELAY_40us:
        PUSH     01H
        MOV      R1,#14H
CONT:
        DJNZ     R1,CONT
        POP      01H
        RET
;--------------------------
DELAY_2ms:
        PUSH     02H
        MOV      R2,#50D
CONL:
        LCALL    DELAY_40us
        DJNZ     R2,CONL
        POP      02H
        RET
;---------------------------
```

```
DELAY_10MS:
          PUSH     03H
          MOV      R3,#05H
DEL:
          LCALL    DELAY_2ms
          DJNZ     R3,DEL
          POP      03H
          RET
DELAY_1S:
          PUSH     00H
          MOV      R0,#64H
DLLY:
          LCALL    DELAY_10MS
          DJNZ     R0,DLLY
          POP      00H
          RET
TAB1:  ;第二屏--以下 IC 字库代码
DB 20H,5AH,48H,4FH,4EH,47H,20H,4CH,49H,20H,20H,31H,36H,2AH,32H,20H
DB 28H,23H,24H,25H,26H,2BH,2DH,2FH,3FH,3DH,5CH,3CH,7EH,7FH,3EH,29H
DB 20H,5AH,48H,4FH,4EH,47H,20H,4CH,49H,20H,20H,31H,36H,2AH,32H,20H
DB 28H,23H,24H,25H,26H,2BH,2DH,2FH,3FH,3DH,5CH,3CH,7EH,7FH,3EH,29H

TAB2:   ;;第三屏--以下 IC 字库代码
DB 28H,0B1H,0B2H,0B3H,0B4H,0B5H,0B6H,0B7H,0B8H,0B9H,0BAH,0BBH,0BCH,0BDH,0BEH,029H
DB 28H,0F1H,0F2H,0F3H,0F4H,0F5H,0F6H,0F7H,0F8H,0F9H,0FAH,0FBH,0FCH,0FDH,0FEH,029H
DB 28H,0B1H,0B2H,0B3H,0B4H,0B5H,0B6H,0B7H,0B8H,0B9H,0BAH,0BBH,0BCH,0BDH,0BEH,029H
DB 28H,0F1H,0F2H,0F3H,0F4H,0F5H,0F6H,0F7H,0F8H,0F9H,0FAH,0FBH,0FCH,0FDH,0FEH,029H

                             END
```

制订：PETER.　　　日期：2007 年 7 月.　　　版本：A.